IN THE CLAIMS:

The following claims replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method of allocating [[a]] processing resources to functions in a queue waiting to be executed, comprising ~~the steps of~~:

determining an amount of ~~the processor resource~~ processing resources available to be assigned within a processor to the functions;

determining an estimate of an amount of the ~~resource~~ processing resources needed for each function waiting in ~~the~~ a queue to execute; and

allocating the available ~~resource~~ processing resources within said processor to the functions based on ~~a hierarchical priority~~ an allocation scheme.

2. (Currently Amended) The method of claim 1, wherein[[:]] the functions are decomposed elements of ~~a more complex process and do not require the same amount of resource to execute~~ one or more algorithms, and

the functions allow management of their computational requirements.

3. (Currently Amended) The method of claim 2, wherein[[:]] multiple ~~instances~~ functions ~~of any function~~ within the ~~process~~ one or more algorithms may be ~~invoked~~ managed by ~~the processor~~ to execute concurrently in the processor.

4. (Currently Amended) The method of claim [[3]] 1, wherein [[:]] said allocating comprises assigning each of the functions ~~within the process is assigned~~ a separate priority within ~~the~~ a hierarchical priority scheme.

5. (Currently Amended) The method of claim [[4]] 1, wherein [[:]] ~~each instance of~~ each function ~~within the process~~ is assigned a separate priority within ~~the hierarchical priority~~ a round-robin allocation scheme.

6. (Currently Amended) The method of claim 2, further comprising ~~the steps of~~:

assigning a resource throttling value to each function waiting in the queue to be executed, wherein the throttling value determines ~~the~~ a reduction of the ~~resource~~ processing resources allocated to each of the functions.

7. (Currently Amended) The method of claim 1, wherein [[:]] the allocation of the available ~~resource~~ processing resources to the functions waiting in the queue is conducted to optimize the amount of the ~~resource~~ processing resources assigned to ~~these~~ the functions.

8. (Canceled)

9. (Currently Amended) The method of claim 1, further comprising ~~the steps of~~:

measuring the actual amount of ~~resource~~ processing resources used by each function within said processor;

determining a revised ~~revising the~~ estimate of the amount of the ~~resource~~ processing resources needed for each function waiting in the queue to execute based on the measured amount of the ~~resource~~ processing resources used; and

reallocating the available amount of the ~~resource~~ processing resources to the functions in accordance with the revised estimate ~~and the hierarchical priority scheme~~.

10. (Currently Amended) The method of claim 9, further comprising ~~the steps of~~:

comparing the measured amount of the ~~resource~~ processing resources used to a high ~~and a low~~ threshold value; and

~~setting~~ executing an alarm if the measured amount of the ~~resource~~ processing resources used exceeds the high threshold value~~; and~~

~~removing the alarm if the measured amount of the resource used is less than the low threshold value~~.

11-16. (Cancelled)

17. (Currently Amended) A method of allocating [[a]] processing resources from a processor to ~~functions~~ instances in a queue waiting to be executed, comprising ~~the steps of~~:

determining an amount of the ~~processor resource~~ processing resources available to be assigned in a processor to a plurality of software instances;

receiving a plurality of communication links into said processor, wherein each said communication link creates a separate instance for said processor to execute;

~~for each of j instances of k functions of j instances, calculating a product obtained by:~~

~~(a)~~ estimating the amount of ~~processor resource~~ processing resources needed to support ~~the~~ execution of the ~~jth instance of the kth function~~ at least one function of each instance in a time period t; and

~~(b) assigning a value of either zero or one to a multiplicand associated with the jth instance of the kth function; and~~

~~(c) multiplying the estimated amount of resource needed to support the execution of the jth instance of the kth function by its associated multiplicand and assigning the result to the product associated with the jth instance of the kth function;~~

~~for each of the j instances, calculating a sub-total sum obtained by:~~

~~(d) summing together the products associated with each of the k functions of the jth instance; and~~

~~(e) adding an estimate of the resource needed to support background processing associated with the jth instance to the sum of the products associated with each of the k functions of the jth instance and assigning the result to the sub-total for the jth instance; and~~

~~allocating the available resource to the k functions of the j instances based on a hierarchical priority scheme~~

if the estimated processing resources exceed the processing resources available in the allocating, then allocating the processing resources in the processor to each function to execute according to an allocation scheme.

18. (Currently Amended) The method of claim 17, wherein [[:]] the ~~multiplicand value~~ allocating ~~associated with the jth instance of the kth function~~ is determined according to ~~the~~ a hierarchical priority scheme.

19. (Canceled)

20. (Currently Amended) The method of claim 19, wherein [[:]] the ~~length~~ allocating comprises executing each function according to a round-robin scheme ~~of each in time period is variable and is no longer than the period needed to execute any one of the j instances of the k functions that are executing concurrently~~.

21. (Currently Amended) The method of claim [[19]] 18, further comprising ~~the step of~~:

for each ~~of the j instances of the kth each~~ instance, assigning an increasingly higher priority in accordance with an increasingly greater number of time periods ~~that have~~ passed since ~~the jth instance of the kth function was last executed~~ execution of a function waiting in said queue.

22. (Currently Amended) The method of claim 17, further comprising ~~the steps of~~:

measuring the actual amount of the ~~resource~~ said processing resources used for said execution of each function;

revising the estimate of the amount of the resource needed for each function waiting in the queue to execute based on the measured amount of the resource used; and

reallocating the available amount of the ~~resource~~ said processing resources to the ~~instances of each function~~ functions in accordance with the revised estimate ~~and the hierarchical priority scheme~~.

23. (Currently Amended) The method of claim 22, further comprising ~~the steps of~~:

comparing the measured amount of ~~the resource~~ said processing resources used to a high ~~and a low~~ threshold value; and

~~setting~~ executing an alarm if the measured amount of ~~the resource~~ said processing resources used exceeds the high threshold value~~; and~~

~~removing the alarm if the measured amount of the resource used is less than the low threshold value~~.

24. (Currently Amended) The method of claim 23, further comprising ~~the step of~~:

assigning a resource throttling value to ~~each instance of~~ each function waiting in the queue to be executed when the alarm is ~~set~~ executed, wherein the throttling value determines the reduction of the ~~resource~~ processing resources allocated to ~~each instance of~~ each of the functions.

25. (Currently Amended) The method of claim 23, further comprising ~~the step of~~:

reducing the number of functions that ~~number of instances in which a particular function~~ may execute concurrently when the alarm is ~~set~~ executed.

26. (New) The method of claim 1, wherein said determining comprises estimating said amount of processing resources that do not exceed a total processing resources of said processor.

27. (New) The method of claim 1, further comprising:

controlling each function to execute within an allocated processing resource in said processor according to said allocation scheme; and

controlling each function to cease execution according to a degradation scheme if a sum said function executions exceed a total processing capacity of said processor that is allocated to said algorithm.

28. (New) A method of resource management in a real-time embedded system, comprising:

estimating processing resource consumption within a processor of each discrete part of an algorithm;

assigning an allocation of processing resources according to said estimate, to said discreet parts of said algorithm, wherein said discrete parts allow management of their execution; and

executing one or more discrete parts in a queueing scheme according to said allocation.

29. (New) The method of claim 28, wherein said estimating processing resources comprises estimates of MIPS (millions of instructions per second) consumption of each function, and further comprises:

updating said consumption MIPS estimations when a state of each function

changes.

30. (New) The method of claim 28, wherein said assigning an allocation of processing resources, within a processor, further comprises:

determining a total MIPS available in said processor available for processing; and

assigning an allocation of MIPS for execution of each of said discrete parts according.

31. (New) The method of claim 28, further comprising:

reducing said allocation if a collective execution of said discrete parts in a time period consume processing resources greater than a threshold set in said processor,

wherein said reducing comprises performing a degradation scheme on one or more discrete parts.

32. (New) The method of claim 17, wherein said estimating processing resources comprises estimates of MIPS (millions of instructions per second) consumption of each function, and further comprises:

updating said consumption MIPS estimations when a state of each function changes.

33. (New) The method of claim 17, wherein said determining comprises estimating said amount of processing resources that do not exceed a total processing resources of said processor.

34. (New) The method of claim 1, wherein said determining an amount of processing resources comprises determining said processing resources for functions of one or more adaptive algorithms, wherein said one or more adaptive algorithms allow modification of their computational requirements.